

学校编码: 10384

分类号 _____ 密级 _____

学号: X2013232340

UDC _____

厦 门 大 学

工 程 硕 士 学 位 论 文

酒店产品推介系统跨平台接口的
设计与实现

Design and Implementation of Multi-Platform Interface
System for Hotel Product Management and Introduce
System

林煜岫

指 导 教 师: 董 槐 林 教 授

专 业 名 称: 软 件 工 程

论文提交日期: 2016 年 09 月

论文答辩日期: 2016 年 11 月

学位授予日期: 2016 年 12 月

指 导 教 师: _____

答辩委员会主席: _____

2016 年 09 月

厦门大学学位论文原创性声明

本人呈交的学位论文是本人在导师指导下,独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果,均在文中以适当方式明确标明,并符合法律规范和《厦门大学研究生学术活动规范(试行)》。

另外,该学位论文为()课题(组)的研究成果,获得()课题(组)经费或实验室的资助,在()实验室完成。(请在以上括号内填写课题或课题组负责人或实验室名称,未有此项声明内容的,可以不作特别声明。)

声明人(签名):

年 月 日

厦门大学学位论文著作权使用声明

本人同意厦门大学根据《中华人民共和国学位条例暂行实施办法》等规定保留和使用此学位论文，并向主管部门或其指定机构送交学位论文（包括纸质版和电子版），允许学位论文进入厦门大学图书馆及其数据库被查阅、借阅。本人同意厦门大学将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于：

（ ） 1.经厦门大学保密委员会审查核定的保密学位论文，
于 年 月 日解密，解密后适用上述授权。

（ ☒ ） 2.不保密，适用上述授权。

（请在以上相应括号内打“√”或填上相应内容。保密学位论文应是已经厦门大学保密委员会审定过的学位论文，未经厦门大学保密委员会审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权。）

声明人（签名）：

年 月 日

摘要

在线旅游业是近十几年来兴起的一个行业。近几年受互联网行业迅速兴起的影响，在线旅游业的业务也呈现飞速发展的态势。为了实现盈利的最大化，各个在线旅游服务商之间开始从竞争转向合作，通过优势产品的共享，实现销量最大化。为了达到这个目的，需要各个服务商之间可以高效准确的实现产品数据共享，最初的通过业务人员手工发送数据的方式已经无法满足日渐增长业务量需求，于是，服务商之间开始考虑通过系统间对接的方式实现数据共享，这样不仅准确性和效率大大提高，实时性也大大增强。甚至，在一些复杂的业务规则约束下的产品价格也能迅速准确地通过系统计算之后对接给对接方，对业务量的提升做出了很大的贡献。

目前，市场上的在线旅游服务提供商数量较多，规模大小参差不齐，业务流程也不尽相同，主要提供的服务包括机票、酒店和门票等。其中，酒店相关的业务规模最庞大，涉及到的服务和业务流程也相对比较复杂。各个酒店集团和分销平台之间有复杂的业务合作关系，一部分核心业务要通过系统对接的方式进行。

所以要实现上述的目标，需要考虑的业务因素和技术因素很多，且都比较复杂。系统的对接首先要在业务层面确定对外开放的数据范围和功能范围，其次，在技术层面确定采用的技术、数据格式和系统对外接口的架构。最后，由于对外开放的接口会提供给多个对接商，所以必须考虑技术的通用性和数据的安全性、一致性和可读性。

本文主要介绍线上旅游服务商之间的产品系统对接的实施方案，包括需求分析和设计实现。该实施方案称为产品系统标准接口，主要采用的技术为 WebService，因为 WebService 技术有标准的传输协议和数据传输格式，且接口描述对各个对接方均可见。服务提供方通过对外开放服务的方式向对接方提供数据访问的入口，并接受对接方的预定数据。

关键词：跨系统对接；接口；WebService

Abstract

Online travel is an emerging industry, it was greatly affected by the rapid development of Internet. Now, the technology which supports the online travel industry is developing rapidly. To maximize the sales volume, most of the online travel agents (OTAs) are seeking for cooperation rather than combination. The cooperation archived by share the products those have obvious advantage effectively and accurately, but the traditional way of sharing product data is manually by the product managers, this can't satisfy the growth of business. So these OTAs are trying to change the sharing way, turn to online auto sharing data via cross-system interface. Online auto sharing would increase the efficiency and accuracy of data transform, even under some complicated business scenarios, the sharing data is calculated by product management system and transform by the cross-system interface, this can avoid errors and promote the sales volume.

Currently, there are a lot of online travel agents in the market. They have different scales and business skills, operating under the different business rules and workflows. These agents provide all kinds of services, including air ticket, hotel reservation and other tickets. Hotel reservation is the most complicated service, this cause the tight cooperation between hotel group and distribution platform, and the core business processes should be proceed by system integrate.

To archive the target of online cross-system data sharing, there are a lot of technical factors should be considered, some of these factor might be very complicated. At first, the open scope of function and business for different distributor should be confirmed by business stuff; secondly, used technology, data transform format and cross-system interface architecture should be confirmed by IT experts and developers; the last but not the least, the universality of adopted technology, the security of data transformation, and the consistency and readability of data should be carefully considered because of the diversity among the multi distributors.

This dissertation mainly introduces the solution of cross-system data sharing interface between online travel agent and its distributors. Including different phases of system implementation, such as requirement analysis, functional design and detail design. WebService was adopted to implement this interface system because of its maturity and universality. WebService has standard data transformation

protocol—SOAP (Simple Object Access Protocol) and data transformation format-XML(Extensible Markup Language). The interface of web services are defined in WSDL (Web Services Description Language) files and published on the internet, it is visible for all the service subscribers (distributors). Online travel agent sharing product data with its distributors by publish web services and accept reservation requests.

Key Word: Cross-System Data Sharing; Interface; WebService

目录

第一章	引言	1
1.1	项目背景	1
1.2	国内外研究现状	2
1.3	本文的主要内容及结构	3
第二章	相关技术介绍	4
2.1	Web Service	4
2.2	CXF 和 Thrift 框架	5
2.3	本章小结	6
第三章	系统需求分析	7
3.1	业务需求	7
3.1.1	业务需求描述	7
3.1.2	业务流程描述	8
3.2	系统功能需求	9
3.2.1	功能描述	9
3.2.2	系统功能与业务需求对应关系	15
3.3	系统非功能需求	15
3.3.1	并发量	15
3.3.2	平均响应时间	16
3.3.3	数据格式	16
3.4	本章小结	16
第四章	总体设计	17
4.1	设计原则与目标	17
4.1.1	设计原则	17
4.1.2	设计目标	18
4.2	架构设计	18
4.2.1	系统层次结构划分	18
4.2.2	WebService 架构设计	19
4.3	系统功能设计	20

4.3.1	价格及库存量接口	21
4.3.2	订单处理接口	22
4.3.3	订单查询接口	23
4.3.4	订单取消接口	23
4.3.5	可定检查接口	24
4.3.6	变价酒店查询接口	25
4.3.7	基础数据获取接口	26
4.4	数据传输格式设计	26
4.5	本章小结	28
第五章	系统详细设计及实现	29
5.1	功能模块详细设计	29
5.2	接口设计	33
5.3	数据格式详细设计	37
5.4	功能模块消息示例	39
5.5	本章小结	49
第六章	总结与展望	50
6.1	总结	50
6.2	展望	50
参考文献	52
致谢	54

Contents

Chapter1 Introduction.....	1
1.1 Research Background.....	1
1.2 Overseas and Domestic Research Status.....	2
1.3 Contents and Structure.....	3
Chapter2 Overview of the Relevent Technologies	4
2.1 Web Service	4
2.2 CXF and Thrift.....	5
2.3 Summary.....	6
Charpter3 System Requirement Analysis	7
3.1 Busniess Requirement	7
3.1.1 Main Business Requirement Description.....	7
3.1.2 Business Workflow Description.....	8
3.2 System Functional Requirement	9
3.2.1 Function Description.....	9
3.2.2 Matchup between Function and Business Requirement	15
3.3 System Non-Functional Requirement	15
3.3.1 Concurrency	15
3.3.2 Average Response Time.....	16
3.3.3 Data Format	16
3.4 Summary.....	16
Chrpter4 System Overall Design.....	17
4.1 System Design Principles and Targets.....	17
4.1.1 Design Principles	17
4.1.2 DesignTargets.....	18
4.2 System Architecture Design	18
4.2.1 System Layers Divide	18
4.2.2 Webservice Architecture Design	19
4.3 System Functional Design	20
4.3.1 Rate and Allotment API	21
4.3.2 Reservation API	22
4.3.3 Query Order API.....	23
4.3.4 Cancel Order API.....	23

4.3.5	Availability API	24
4.3.6	Hotel Change API	25
4.3.7	Hotel Basic Information API	26
4.4	Data Format Design	26
4.5	Summary	28
Chapter5 System Detail Design and Implementation		29
5.1	Function Modules Detail Design	29
5.2	Interface Design	33
5.3	Data Format Detail Design	37
5.4	SOAP Messages Demostration	39
5.5	Summary	49
Chapter6 Conclusions and Prospect		50
6.1	Conclusions	50
6.2	Prospect	50
References		52
Acknowledgements		54

第一章 引言

1.1 项目背景

近年来由于社会经济的飞速发展，群众的物质水平有了大幅提高，对精神生活的追求也不断提高，旅游业开始迅速发展起来。在互联网行业飞速发展的带动下，线上旅游也开始快速发展起来，对于线上旅游业的技术支撑的要求也越来越高。线上旅游的技术需要解决高并发量、大数据量以及多系统跨平台对接的问题。

多系统跨平台对接是指多个在线旅游服务商之间的产品数据管理系统的交互，通过系统之间对接的方式实现产品数据的共享，业务层面的意义是实现优势互补和资源共享，提供更好的服务和产品给消费者。

目前，市场上的在线旅游服务提供商数量较多，规模大小参差不齐，业务流程也不尽相同，主要提供的服务包括机票、酒店和门票等。其中，酒店相关的业务规模最庞大，涉及到的服务和业务流程也相对比较复杂。各个酒店集团和分销平台之间有复杂的业务合作关系，一部分核心业务要通过系统对接的方式进行。

所以要实现上述的目标，需要考虑的业务因素和技术因素很多，且都比较复杂。系统的对接首先要在业务层面确定对外开放的数据范围和功能范围，其次在技术层面确定采用的技术、数据格式和系统对外接口的架构。由于对外开放的接口会提供给多个对接商，所以必须考虑技术的通用性和数据的安全性、一致性和可读性。而且，为了业务发展的需要，针对某些特定的对接商，接口还需要做出一定的调整，所以对接口架构的灵活性要求也很高。必须保证在不影响接口原有功能和数据结构的基础上做调整。

基于以上的考虑，基于 SOAP 协议的 WebService 技术是目前较优的一种选择。WebService 技术的核心实际就是服务方提供一个可使用 SOAP 消息通过网络来访问的接口，这个接口以 WSDL 文档的形式描述了一组可访问的操作，供外界调用。跨平台、通用性强、可扩展性强，可以提供一种松散的、分布式的信息交换实现机制，这些都是 WebService 的优势。因此，在技术层面，采用 WebService 技术完全可以满足系统对接的要求。

1.2 国内外研究现状

对于 WebService 的定义, W3C Web Service Architecture 小组给出的一致意见如下: WebService 是一种可以通过 XML 定义、描述、发现其接口的应用程序, 它是由 URI 标识的。WebService 能够支持基于 XML 的消息通过 Internet 协议跟其他应用程序直接进行交互。在 WebService 有四个关键技术环节: WSDL、SOAP、XML 和 UDDI。其中, WSDL 用于描述服务, 通过 SOAP 协议来调用服务和传输数据, 使用 XML 格式来编码数据, 借由 UDDI 来发布和查找服务。

WebService 使 Web 应用程序上升到另一高度, 通过使用 WebService, 应用程序可以向任何你希望的调用者发布功能或消息^[6]。W3C 将 WebService 的应用类型分为两种,

- 可重复使用的应用程序组件——WebService 可以把应用程序的组件作为服务来提供, 达到组件重复使用的目的。
- 和连接现有的软件——WebService 可以为不同的应用程序提供一种链接其数据的途径, 达到在不同平台之间交换数据的目的。

随着 WebService 技术的广泛应用, 软件行业内一些影响力比较大的公司, 如微软、IBM 等, 联合起来制定了一系列 WebService 相关的规范和标准, 如 WS-Security。这些规范和标准为 WebService 技术的推广和进一步完善做出了很大的贡献。

目前, WebService 的实现方案已经发展出很多种, 主流有 SOAP、REST 和 XML-RPC 三种, 其中 REST (表征状态转移 Representational State Transfer) 是由 Roy Fielding 在 2000 年时他的博士论文中首次提出, 其风格比 SOAP 简洁, 虽然成熟度不如 SOAP 高, 但胜在效率和易用性上。已经有越来越多的大型软件公司开始转向 REST, 如 Amazon.com 和雅虎等公司提供的 Web 服务都采用的 REST 风格。

在国内, 随着互联网行业的迅速兴起, 许多企业为了发展, 必须要打破企业内部信息孤岛的限制, 也要跟行业内的其他公司达成协作关系。Web 服务的采用几乎已经成了企业生死攸关的选择, 国内很多大型的 IT 公司甚至开始推出自己的 WebService 产品, 例如, 阿里巴巴旗下的子公司阿里云推出的同名产品——阿里云等。

1.3 本文的主要内容及结构

本文详细描述一个基于 WebService 的适用性强、效率高的酒店产品系统对接接口的设计和实现过程。此接口用于实现不同线上旅游服务商之间可以通过少量的开发和配置工作实现系统对接的目的。

本项目采用 JAVA 语言实现，应用的技术为 WebService，针对各个服务商之间的平台差异和技术差异，设计并实现一套通用性强、效率高的标准接口，部署本接口之后，对接的系统可以通过接口获取产品数据和推送订单数据。并可以通过日志来记录和跟踪对接的各个渠道商的订单量，方便业务人员评估渠道商质量。

本课题研究将集中在如何设计和实现标准接口上，如何应用当前主流的通用的软件技术辅以业务规则，实现数据格式在技术和业务两个层面的标准化。根据业务需求，标准接口由“酒店基本信息查询”，“酒店价格查询”和“酒店预订”三大块主要功能构成。对外提供统一的数据格式。目前 WebService 技术的应用已经比较成熟，本课题的实施主要关注在业务方面为企业带来的意义。

本文共六章，组织结构如下：

第一章主要描述项目背景和采用的技术，介绍所用的技术在当前国内外研究和应用现状。

第二章针对本项目应用的主要技术，进行了简单的介绍。

第三章重点介绍项目的需求分析，描述项目的业务流程，根据业务需要来分析系统的功能需求和非功能需求。

第四章将详细介绍项目的设计过程，介绍系统的设计原则，在设计原则的指导下进行架构设计和功能设计，实现第二章中描述的需求。

第五章主要描述项目的详细设计内容，包括系统各个功能模块的详细设计和系统对内对外接口的详细设计。

第六章将对项目实施过程中解决的问题和遇到的问题进行汇总，并展望下一步的工作。

第二章 相关技术介绍

本章主要介绍系统采用的技术，包括发布对外接口的 Web Service 技术，生成 WSDL 文件的 CXF 框架和生成内部接口代码的 Thrift 框架。

2.1 Web Service

Web Service 技术主要用于本系统发布服务，向外系统提供访问本系统功能的入口。

Web service 是一个平台独立的，低耦合的，自包含的、基于可编程的 web 的应用程序，可使用开放的 XML（标准通用标记语言下的一个子集）标准来描述、发布、发现、协调和配置这些应用程序，用于开发分布式的互操作的应用程序。

Web Service 平台需要一套协议来实现分布式应用程序的创建。任何平台都有它的数据表示方法和类型系统。要实现互操作性，Web Service 平台必须提供一套标准的类型系统，用于沟通不同平台、编程语言和组件模型中的不同类型系统。这些协议有：

XML 和 XSD，可扩展的标记语言（标准通用标记语言下的一个子集）是 Web Service 平台中表示数据的基本格式。除了易于建立和易于分析外，XML 主要的优点在于它既与平台无关，又与厂商无关。XML 是由万维网协会(W3C)创建，W3C 制定的 XML Schema XSD 定义了一套标准的数据类型，并给出了一种语言来扩展这套数据类型。

Web Service 平台是用 XSD 来作为数据类型系统的。当你用某种语言如 VB，.NET 或 C#来构造一个 Web Service 时，为了符合 Web Service 标准，所有你使用的数据类型都必须被转换为 XSD 类型。如想让它使用在不同平台和不同软件的不同组织间传递，还需要用某种东西将它包装起来。这种东西就是一种协议，如 SOAP。

SOAP 即简单对象访问协议(Simple Object Access Protocol)，它是用于交换 XML（标准通用标记语言下的一个子集）编码信息的轻量级协议。它有三个主

要方面：XML-envelope 为描述信息内容和如何处理内容定义了框架，将程序对象编码成为 XML 对象的规则，执行远程过程调用(RPC)的约定。SOAP 可以运行在任何其他传输协议上。例如，你可以使用 SMTP，即因特网电子邮件协议来传递 SOAP 消息，这可是很有诱惑力的。在传输层之间的头是不同的，但 XML 有效负载保持相同。

Web Service 希望实现不同的系统之间能够用“软件-软件对话”的方式相互调用，打破了软件应用、网站和各种设备之间的格格不入的状态，实现“基于 Web 无缝集成”的目标。

WSDL，Web Service 描述语言 WSDL 就是用机器能阅读的方式提供的一个正式描述文档而基于 XML（标准通用标记语言下的一个子集）的语言，用于描述 Web Service 及其函数、参数和返回值。因为是基于 XML 的，所以 WSDL 既是机器可阅读的，又是人可阅读的。

UDDI 的目的是为电子商务建立标准；UDDI 是一套基于 Web 的、分布式的、为 Web Service 提供的、信息注册中心的实现标准规范，同时也包含一组使企业能将自身提供的 Web Service 注册，以使别的企业能够发现的访问协议的实现标准。

调用 RPC 与消息传递，Web Service 本身其实是在实现应用程序间的通信。我们有两种应用程序通信的方法：RPC 远程过程调用 和消息传递。使用 RPC 的时候，客户端的概念是调用服务器上的远程过程，通常方式为实例化一个远程对象并调用其方法和属性。RPC 系统试图达到一种位置上的透明性：服务器暴露出远程对象的接口，而客户端就好像在本地使用的这些对象的接口一样，这样就隐藏了底层的信息，客户端也就根本不需要知道对象是在哪台机器上。

2.2 CXF 和 Thrift 框架

Apache CXF 框架在本系统中的应用包括两部分，第一部分是用于生成本系统所有 Web Service 的 WSDL 文件，第二部分是用于将描述 POJO 对象的 XSD 文件转换成 JAVA 类文件。

Apache CXF 是一个开源的 Services 框架，CXF 帮助开发人员利用 Frontend 编程 API 来构建和开发 Services，像 JAX-WS。这些 Services 可以支持多种协议，

比如：SOAP、XML/HTTP、RESTful HTTP 或者 CORBA，并且可以在多种传输协议上运行，比如：HTTP、JMS 或者 JBI，CXF 大大简化了 Services 的创建，同时它继承了 XFire 传统，一样可以天然地和 Spring 进行无缝集成。

CXF 特点：

- 轻量级容器，灵活部署
- 支持多种编程语言
- 支持多种代码生成方式

Thrift 用于生成本系统与产品管理系统间的接口代码。

Thrift 是一个软件框架，用来进行可扩展且跨语言的服务的开发。它结合了功能强大的软件堆栈和代码生成引擎，以构建在 C++, Java, Go, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, JavaScript, Node.js 和 Smalltalk 等编程语言间无缝结合的、高效的服务。

2.3 本章小结

本章针对接口系统主要应用的技术进行了概要性的介绍，目的是明确系统的主要技术路线。

Degree papers are in the “[Xiamen University Electronic Theses and Dissertations Database](#)”.

Fulltexts are available in the following ways:

1. If your library is a CALIS member libraries, please log on <http://etd.calis.edu.cn/> and submit requests online, or consult the interlibrary loan department in your library.
2. For users of non-CALIS member libraries, please mail to etd@xmu.edu.cn for delivery details.